

# Error Performance Analysis of the Symbol-Decision SC Polar Decoder

Chenrong Xiong, Jun Lin, *Student member, IEEE* and Zhiyuan Yan, *Senior member, IEEE*

**Abstract**—Polar codes are the first provably capacity-achieving forward error correction codes. To improve decoder throughput, the symbol-decision SC algorithm makes hard-decision for multiple bits at a time. In this paper, we prove that for polar codes, the symbol-decision SC algorithm is better than the bit-decision SC algorithm in terms of the frame error rate (FER) performance because the symbol-decision SC algorithm performs a *local* maximum likelihood decoding within a symbol. Moreover, the bigger the symbol size, the better the FER performance. Finally, simulation results over both the additive white Gaussian noise channel and the binary erasure channel confirm our theoretical analysis.

**Index Terms**—Error control codes, polar codes, successive cancellation, performance analysis, bit-decision decoding, symbol-decision decoding

## I. INTRODUCTION

Polar codes, a groundbreaking discovery by Arikan [1], provably achieve the capacity of any discrete [1] and continuous [2] memoryless channels. Since their debut, a lot of efforts have been made to improve the error performance of short polar codes. Although a sphere decoding algorithm [3], stack sphere decoding algorithm [4] or a Viterbi algorithm [5] can provide maximum likelihood (ML) decoding of polar codes, they are considered infeasible due to their high complexity. Compared with these ML decoding algorithms, the successive cancellation (SC) decoding algorithm [1] has a lower complexity at the cost of sub-optimal performance. Another drawback of the SC algorithm is its long decoding latency and low decoding throughput because the SC algorithm makes hard *bit* decisions only one bit at a time. To reduce the decoding latency and improve the throughput from a perspective of parallel processing, a parallel SC algorithm was proposed in [6] and [7]. The symbol-decision SC algorithm in [8], which has the same decoding schedule as the parallel SC algorithm, makes hard *symbol* decisions one at a time. In terms of error performance, numerical simulation results in [6] and [7] were used to show that the parallel SC algorithm has no performance loss compared with the SC algorithm. There is no theoretical analysis in the literature that shows whether the parallel and symbol-decision SC algorithms are superior or inferior to the SC algorithm [1], referred to as the bit-decision SC algorithm henceforth.

In this paper, besides numerical simulations, we prove that in terms of frame error rate (FER) performance, the symbol-decision SC algorithm is better than the bit-decision SC algorithm. Moreover, the bigger the symbol size, the better the FER performance. Finally, simulation results over the additive

white Gaussian noise (AWGN) channel and the binary erasure channel (BEC) confirm our theoretical analysis.

The rest of this paper is organized as follows. In Section II, polar codes are reviewed as well as the bit- and symbol-decision SC algorithms. In Section III, we prove that the symbol-decision SC algorithm has a better FER performance than the bit-decision SC algorithm. In this section, we also show how to make use of future frozen bits within a symbol by the symbol-decision SC algorithm. Numerical simulation results are presented to confirm our theoretical conclusion as well. Finally, some conclusions are provided in Section IV.

## II. BIT-DECISION AND SYMBOL-DECISION SC ALGORITHMS FOR POLAR CODES

### A. Polar codes

For simplicity, we denote  $(u_a, u_{a+1}, \dots, u_{b-1}, u_b)$  as  $u_a^b$ ; if  $a > b$ ,  $u_a^b$  is regarded as void. For any index set  $\mathcal{A} \subseteq \mathcal{I} = \{1, 2, \dots, N\}$ ,  $\mathbf{u}_{\mathcal{A}} = (u_i : 0 < i \leq N, i \in \mathcal{A})$  is the sub-sequence of  $\mathbf{u} = u_1^N$  restricted to  $\mathcal{A}$ . We denote the complement of  $\mathcal{A}$  in  $\mathcal{I}$  as  $\mathcal{A}^c$ .

Suppose  $N = 2^n$ , for an  $(N, K)$  polar code, the data bit sequence  $\mathbf{u} = u_1^N$  is divided into two parts: a  $K$ -element part  $\mathbf{u}_{\mathcal{A}}$  which carries information bits, and  $\mathbf{u}_{\mathcal{A}^c}$  whose elements are predefined frozen bits. For convenience, frozen bits are set to zero.

To generate the corresponding encoded bit sequence  $\mathbf{x} = x_1^N$ ,

$$\mathbf{x} = \mathbf{u} B_N F^{\otimes n}, \quad (1)$$

where  $B_N$  is the  $N \times N$  bit-reversal permutation matrix,  $F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ , and  $F^{\otimes n}$  is the  $n$ -th Kronecker power of  $F$  [1].

### B. Bit-Decision SC Algorithm for Polar Codes

When  $\mathbf{x}$  is transmitted, suppose the received bits are  $\mathbf{y} = y_1^N$ . The bit-decision SC algorithm [1] for an  $(N, K)$  polar code estimates the data bit sequence  $\mathbf{u}$  successively: for  $j = 1, 2, \dots, N$ ,  $\hat{u}_j = 0$  if  $u_j$  is a frozen bits, otherwise it is estimated by

$$\hat{u}_j = \arg \max_{u_j \in \{0, 1\}} \Pr(\mathbf{y}, \hat{u}_1^{j-1} | u_j). \quad (2)$$

Here, the bit-decision SC algorithm makes hard bit decisions one bit at a time.

### C. Symbol-Decision SC Algorithm for Polar Codes

The  $M$ -bit<sup>1</sup> parallel and symbol-decision SC algorithms [6]–[8] make hard-decision for  $M$  bits instead of only one bit at a time. For  $0 \leq j < \frac{N}{M}$ , the  $j$ -th symbol is estimated successively by

$$\hat{u}_{jM+1}^{jM+M} = \arg \max_{\substack{\mathbf{u}_{\mathcal{AM}_j} \in \{0,1\}^{|\mathcal{AM}_j|} \\ \mathbf{u}_{\mathcal{AM}_j^c} \in \{0\}^{|\mathcal{AM}_j^c|}}} \Pr(\mathbf{y}, \hat{u}_1^{jM} | u_{jM+1}^{jM+M}), \quad (3)$$

where  $\mathcal{IM}_j \stackrel{\text{def}}{=} \{jM+1, jM+2, \dots, jM+M\} \subseteq \mathcal{I}$ ,  $\mathcal{AM}_j \stackrel{\text{def}}{=} \mathcal{IM}_j \cap \mathcal{A}$ ,  $\mathcal{AM}_j^c \stackrel{\text{def}}{=} \mathcal{IM}_j \cap \mathcal{A}^c$ , and  $|\mathcal{AM}_j|$  represents the cardinality of  $\mathcal{AM}_j$ . If  $M = N$ , the  $M$ -bit symbol-decision SC algorithm is exactly an ML sequence decoding algorithm.

### III. PERFORMANCE ANALYSIS OF THE SYMBOL-DECISION SC ALGORITHM

#### A. FER Analysis of the Symbol-Decision SC Decoding Algorithm

To have a fair comparison, we assume the symbol-decision decoding has the same bit sequence  $\mathbf{u}$  as its bit-decision counterpart. Without loss of generality, we consider two decoding scenarios shown in Fig. 1. In both scenarios, an  $N$ -bit vector is divided into  $\frac{N}{M}$  segments. Each segment has  $M$  bits. The bit-decision SC and  $M$ -bit ML decoding algorithms are used to decode each segment of scenarios (a) and (b), respectively. A box means that a decision is made. From a segment to the following segment, both scenarios use the same schedule – the successive schedule. Then scenarios (a) and (b) exactly correspond to the bit-decision SC and  $M$ -bit symbol-decision SC algorithms, respectively. Note that when a different bit sequence is used for both, all conclusions still apply.

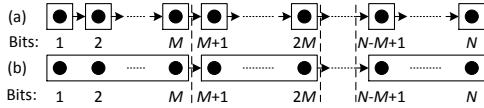


Fig. 1. Decoding procedures of (a) a bit-decision SC algorithm and (b) an  $M$ -bit symbol-decision SC algorithm.

In terms of the FER performance, we have

**Proposition 1.** If all data sequences are independent and equally likely, for an  $(N, K)$  polar code over any given channel, the FER of the bit-decision SC algorithm  $\Pr_B(\hat{u}_1^N \neq u_1^N)$  and the FER of the  $M$ -bit symbol-decision SC algorithm  $\Pr_M(\hat{u}_1^N \neq u_1^N)$  satisfy:

$$\Pr_M(\hat{u}_1^N \neq u_1^N) \leq \Pr_B(\hat{u}_1^N \neq u_1^N). \quad (4)$$

*Proof:* Let us calculate the FERs of the two scenarios shown in Fig. 1. Let  $p_0 = \Pr_{SC}(\hat{u}_1^M \neq u_1^M)$  and  $p'_0 = \Pr_{ML}(\hat{u}_1^M \neq u_1^M)$  denote the segment error rate of  $\hat{u}_1^M \neq u_1^M$  by using the SC and ML decoding algorithms, respectively. Similarly, for  $i = 1, 2, \dots, \frac{N}{M} - 1$ , let

$p_i = \Pr_{SC}(\hat{u}_{iM+1}^{iM+M} \neq u_{iM+1}^{iM+M} | \hat{u}_1^M = u_1^M)$  and  $p'_i = \Pr_{ML}(\hat{u}_{iM+1}^{iM+M} \neq u_{iM+1}^{iM+M} | \hat{u}_1^M = u_1^M)$  ( $1 \leq i < \frac{N}{M}$ ) represent the probabilities of that the  $i$ -th segment is erroneously decoded by the SC and ML decoding algorithms, respectively, provided that all previous segments are correctly decoded.

Then we have the segment error probability  $\Pr(\hat{u}_1^M \neq u_1^M) = \sum_{y_1^N} \Pr(\hat{u}_1^M \neq u_1^M | y_1^N) \Pr(y_1^N)$ . Since  $\Pr(y_1^N)$  is independent of the decoding rule, to minimize  $\Pr(\hat{u}_1^M \neq u_1^M)$ , we need to minimize  $\Pr(\hat{u}_1^M \neq u_1^M | y_1^N)$ , i.e., to maximize  $\Pr(\hat{u}_1^M = u_1^M | y_1^N)$ .

Because

$$\Pr(u_1^M | y_1^N) = \frac{\Pr(y_1^N | u_1^M) \Pr(u_1^M)}{\Pr(y_1^N)},$$

and  $u_1^M$  is a uniformly distributed random variable, the ML decoder maximizes  $\Pr(\hat{u}_1^M = u_1^M | y_1^N)$ . Therefore, we have

$$p_0 \geq p'_0. \quad (5)$$

For any  $1 \leq i < \frac{N}{M}$ , the segment error probability  $\Pr(\hat{u}_{iM+1}^{iM+M} \neq u_{iM+1}^{iM+M} | \hat{u}_1^M = u_1^M) = \sum_{y_1^N} \Pr(\hat{u}_{iM+1}^{iM+M} \neq u_{iM+1}^{iM+M} | y_1^N, \hat{u}_1^M = u_1^M) \Pr(y_1^N)$ . Hence, to minimize  $\Pr(\hat{u}_{iM+1}^{iM+M} \neq u_{iM+1}^{iM+M} | \hat{u}_1^M = u_1^M)$ ,  $\Pr(\hat{u}_{iM+1}^{iM+M} = u_{iM+1}^{iM+M} | y_1^N, \hat{u}_1^M = u_1^M)$  need to be maximized.

Because

$$\Pr(u_{iM+1}^{iM+M} | y_1^N, \hat{u}_1^M = u_1^M) = \frac{\Pr(y_1^N, \hat{u}_1^M = u_1^M | u_{iM+1}^{iM+M}) \Pr(u_{iM+1}^{iM+M})}{\Pr(y_1^N, \hat{u}_1^M = u_1^M)},$$

and  $u_{iM+1}^{iM+M}$  is a uniformly distributed random variable, the ML decoder maximizes  $\Pr(\hat{u}_{iM+1}^{iM+M} = u_{iM+1}^{iM+M} | y_1^N, \hat{u}_1^M = u_1^M)$ . Therefore, we also have

$$p_i \geq p'_i \text{ for } 1 \leq i < \frac{N}{M}. \quad (6)$$

For the bit-decision SC algorithm,

$$\Pr_B(\hat{u}_1^N \neq u_1^N) = 1 - \prod_{i=0}^{\frac{N}{M}-1} (1 - p_i).$$

For the  $M$ -bit symbol-decision SC algorithm,

$$\Pr_M(\hat{u}_1^N \neq u_1^N) = 1 - \prod_{i=0}^{\frac{N}{M}-1} (1 - p'_i).$$

According to (5) and (6), we have

$$\Pr_M(\hat{u}_1^N \neq u_1^N) \leq \Pr_B(\hat{u}_1^N \neq u_1^N).$$

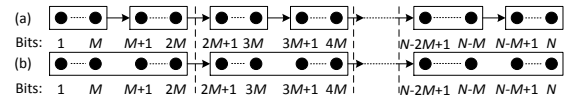


Fig. 2. Decoding procedures of (a) an  $M$ -bit symbol-decision SC algorithm and (b) a  $2M$ -bit symbol-decision SC algorithm.

Furthermore, we have

<sup>1</sup>Although the symbol size  $M$  can be any integer no more than  $N$ , we assume  $M|N$  for simplicity.

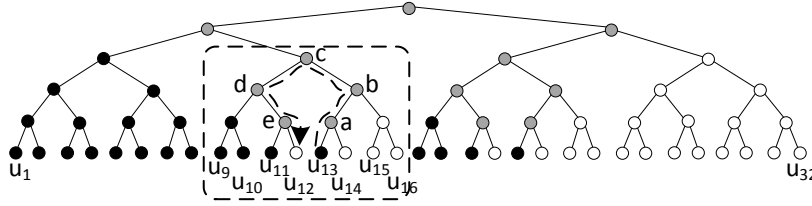


Fig. 3. Tree graph of a (32, 16) polar code.

**Proposition 2.** If all data sequences are independent and equally likely, for an  $(N, K)$  polar code over any given channel, the FER of an  $M$ -bit symbol-decision SC algorithm  $\Pr_M(\hat{u}_1^N \neq u_1^N)$  and the FER of a  $2M$ -bit symbol-decision SC algorithm  $\Pr_{2M}(\hat{u}_1^N \neq u_1^N)$  satisfy:

$$\Pr_{2M}(\hat{u}_1^N \neq u_1^N) \leq \Pr_M(\hat{u}_1^N \neq u_1^N). \quad (7)$$

By considering the two scenarios in Fig. 2, Proposition 2 can be proved in a similar way as for Proposition 1.

Therefore, the symbol-decision SC algorithm is no worse than the bit-decision SC algorithm in terms of the FER performance and bridges the FER performance gap between the bit-decision SC algorithm and the ML decoding algorithm.

### B. Message Passing Interpretation

The SC algorithm can be considered as message passing over a tree graph [9]. From the perspective of message passing over a tree graph, we provide an explanation of the advantage of the symbol-decision decoding. To this end, we introduce a string vector  $\mathbb{S}_i = \mathcal{S}_{i,1}, \dots, \mathcal{S}_{i,M}$  (for  $0 \leq i < \frac{N}{M}$ ) to represent a data pattern of the  $i$ -th  $M$ -bit symbol of a polar code with length  $N$ . If  $u_{iM+j}$  is an information bit,  $\mathcal{S}_{i,j}$  is denoted as 'D'. Otherwise,  $\mathcal{S}_{i,j}$  as 'F'. Consider a toy example of a 4-bit symbol  $u_{4i+4}^{4i+4}$ . Assuming  $u_{4i+1}$  and  $u_{4i+3}$  are information bits, and  $u_{4i+2}$  and  $u_{4i+4}$  are frozen bits. Then the data pattern of  $u_{4i+4}^{4i+4}$  is 'DFDF'. Obviously, for an  $M$ -bit symbol, there are  $2^M$  possible data patterns. We divide them into two types. The first type is called a DP-I pattern, which has no 'D' or has no 'F' after the first 'D'. There are only  $(M+1)$  DP-I patterns. The remaining  $(2^M - M - 1)$  patterns are called DP-II patterns. Henceforth, a symbol which has a DP-I (DP-II, respectively) pattern is called a DP-I (DP-II, respectively) symbol.

As pointed out in [1], the bit-decision decoding does not take advantage of future frozen bits. That is, when decoding information bit  $u_i$  ( $i \in \mathcal{A}$ ), the fact that  $u_j$  ( $j \in \mathcal{A}$  and  $j > i$ ) is a frozen bit is not accounted for by the bit-decision SC algorithm. For the symbol-decision SC algorithm, the future frozen bits in future symbols and within a DP-I symbol cannot be taken advantage of either. However, the decision rule of the symbol-decision SC algorithm can be regarded as a *local* ML decoder. As a result, some information bits can take advantage of their future frozen bit(s) within any DP-II symbol.

We consider a tree graph representation, shown in Fig. 3, of a (32, 16) polar code constructed with the method in [10]. Nodes on the bottom (from left to right,  $u_1$  to  $u_{32}$ ) are called leaf nodes. Each leaf node corresponds to a data bit. There are three kinds of nodes in the tree graph. A rate-0 node whose

descendant leaf nodes are all frozen bits is represented by a black node. A rate-1 node whose descendant leaf nodes are all information bits is represented by a white node. The rest are rate-R nodes in gray. Some descendant leaf nodes of a rate-R node are frozen bits, and the others are information bits. We consider how to use the knowledge of a frozen bit from the perspective of message passing. The knowledge of a frozen bit can be passed through only the rate-0 nodes according to the encoding of polar codes.

Given a tree graph and  $M$ , data patterns are determined. For the tree graph in Fig. 3, all data patterns of  $M = 2, 4$ , and 8 are listed in Table I. When  $M = 2$  and 4, there are no DP-II symbols.

TABLE I  
DATA PATTERNS OF THE (32, 16) POLAR CODE FOR DIFFERENT  $M$ s.

$M$	DP-I	DP-II
2	FF, FD, DD	none
4	FFFF, FFFD, FDDD, DDDD	none
8	FFFFFFFF, DDDDDDDD	FFFD, FDDD

Let us take the decoding of  $u_{12}$  as an example. Although  $u_{13}$  is a frozen bit, this knowledge needs to pass through some intermediate nodes  $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$  before being received by  $u_{12}$  if it is to be taken advantage of in the decoding of  $u_{12}$ . However, because there is at least one rate-R node in the message passing route from  $u_{13}$  to  $u_{12}$ , the decoding of  $u_{12}$  cannot take advantage of the frozen bit  $u_{13}$ . However, for the 8-bit symbol-decision SC algorithm, the DP-II symbol  $u_9^{16}$  is decoded as a symbol simultaneously. The frozen bits ( $u_9^{11}$  and  $u_{13}$ ) help to decode the information bits ( $u_{12}$  and  $u_{14}^{16}$ ). Therefore, unlike the bit-decision SC decoding algorithm, the 8-bit symbol-decision SC decoding does take advantage of  $u_{13}$  to decode  $u_{12}$ . If the 2-bit or 4-bit symbol-decision algorithm are used, no future frozen bits can be taken advantage of in decoding any information bit because all 2-bit or 4-bit symbols are DP-I symbols. In terms of the FER over the BEC, SDSC-32 (ML) < SDSC-16  $\approx$  SDSC-8 < SDSC-4  $\approx$  SDSC-2  $\approx$  SC (shown in Fig. 4), where SDSC- $i$  represents the  $i$ -bit symbol-decision SC algorithm and SDSC-32 is also an ML algorithm.

### C. Simulation Results

Figs. 5 and 6 show bit error rates (BERs) and FERs of symbol-decision SC algorithms with different symbol sizes for a (1024, 512) polar code constructed by the method in [10] over the AWGN channel and the BEC. Regarding data patterns of the (1024, 512) polar code, all 2-bit and 4-bit data symbols are DP-I symbols. However, for the SDSC-8 algorithm, 8

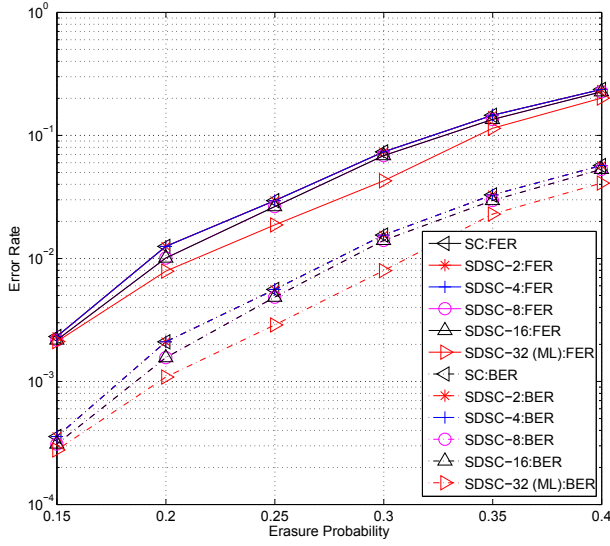


Fig. 4. Error rates of decoding algorithms for the (32, 16) polar code over the BEC.

of 128 data symbols are DP-II symbols. For the SDSC-16 algorithm, 12 of 64 data symbols are DP-II symbols. In terms of the FER,  $\text{SDSC-16} < \text{SDSC-8} < \text{SDSC-4} \approx \text{SDSC-2} \approx \text{SC}$  for the (1024, 512) polar code. The simulation results are consistent with Propositions 1 and 2.

The performance gains are small in our simulation results, but these simulation results still reveal how the symbol size affects the FER performance of the symbol-decision SC algorithm. If a larger performance gain is expected, the symbol size should be increased further. However, for larger symbol sizes, we do not provide the simulation results because simulations are very time-consuming.

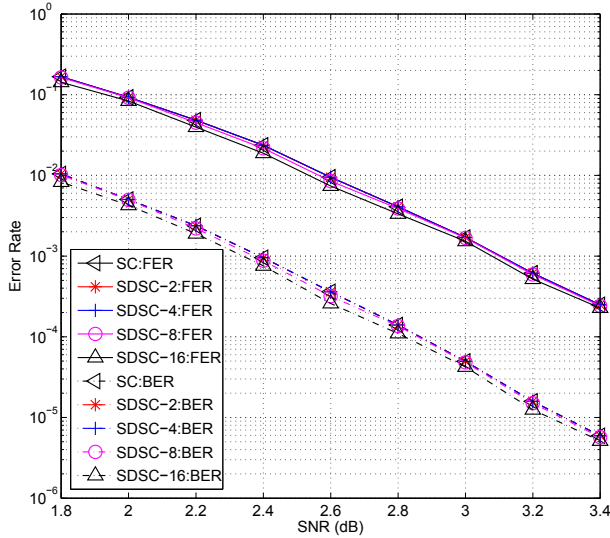


Fig. 5. Error rates of decoding algorithms for the (1024, 512) polar code over the AWGN channel.

In terms of the BER performance, although we cannot offer a rigorous proof, we conjecture that the symbol-decision SC algorithm is better than the bit-decision SC algorithm. The simulation results in Figs. 5 and 6 are consistent with this

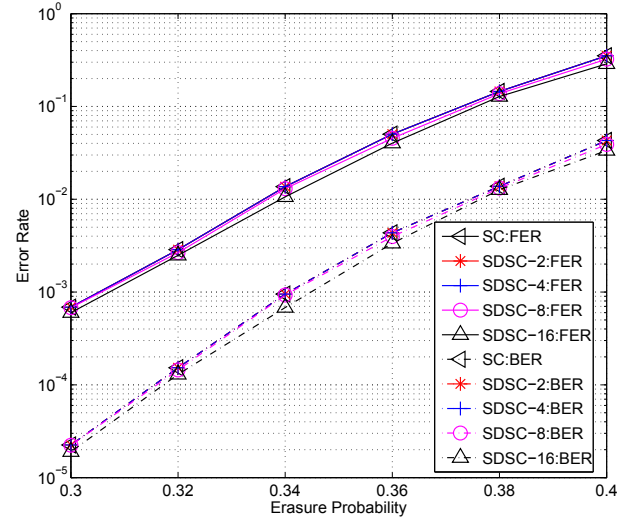


Fig. 6. Error rates of decoding algorithms for the (1024, 512) polar code over the BEC.

conjecture.

#### IV. CONCLUSION

This letter proves that the symbol-decision SC algorithm performs better than the bit-decision SC algorithm for polar codes in terms of the FER performance. Increasing the symbol size increases the FER performance gain. Therefore, the symbol-decision SC algorithm bridges the FER performance gap between the bit-decision SC algorithm and the ML decoding algorithm for polar codes.

#### REFERENCES

- [1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.
- [2] E. Sasoglu, I. Telatar, and E. Arikan, "Polarization for arbitrary discrete memoryless channels," in *Proceedings of IEEE Information Theory Workshop*, Oct 2009, pp. 144–148.
- [3] S. Kahrman and M. Celebi, "Code based efficient maximum-likelihood decoding of short polar codes," in *Proceedings of IEEE International Symposium on Information Theory*, July 2012, pp. 1967–1971.
- [4] K. Niu, K. Chen, and J. Lin, "Low-complexity sphere decoding of polar codes based on optimum path metric," *IEEE Communications Letters*, vol. PP, no. 99, pp. 1–4, 2014.
- [5] E. Arikan, H. Kim, G. Markarian, U. Ozgur, and E. Poyraz, "Performance of short polar codes under ML decoding," in *Proceedings of ICT Mobile Summit Conference*, 2009.
- [6] B. Li, H. Shen, and D. Tse, "Parallel decoders of polar codes," arXiv:1309.1026, September 2013. [Online]. Available: <http://arxiv.org/abs/1309.1026>
- [7] B. Yuan and K. Parhi, "Low-latency successive-cancellation list decoders for polar codes with multibit decision," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–13, 2014.
- [8] C. Xiong, J. Lin, and Z. Yan, "Symbol-based successive cancellation list decoder for polar codes," in *Proceedings of IEEE Workshop on Signal Processing Systems (SiPS 2014)*, Belfast, UK, October 2014, pp. 198–203. [Online]. Available: <http://arxiv.org/abs/1405.4957>
- [9] A. Alamdar-Yazdi and F. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1378–1380, Dec. 2011.
- [10] E. Arikan, "A performance comparison of polar codes and Reed-Muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447–449, June 2008.